arm

BOSTON UNIVERSITY

Customizing Federated Learning to the Edge

Venkatesh Saligrama

Boston University

MLSys Workshop, April 2021

BOSTON UNIVERSITY

Project Team

Acar et.al. ICLR 2021, Federated Learning Based on Dynamic Regularization
Acar et.al. ICML 2021, Personalized Federated Learning Based on Debiasing

Alp Acar[1]

Yue Zhao[2]

Ruizhao Zhu[1]

Ramon Matas Navarro[2]

Matthew Mattina[2]

Paul N. Whatmough[2]

[1]Boston University
[2]Arm ML Research Lab

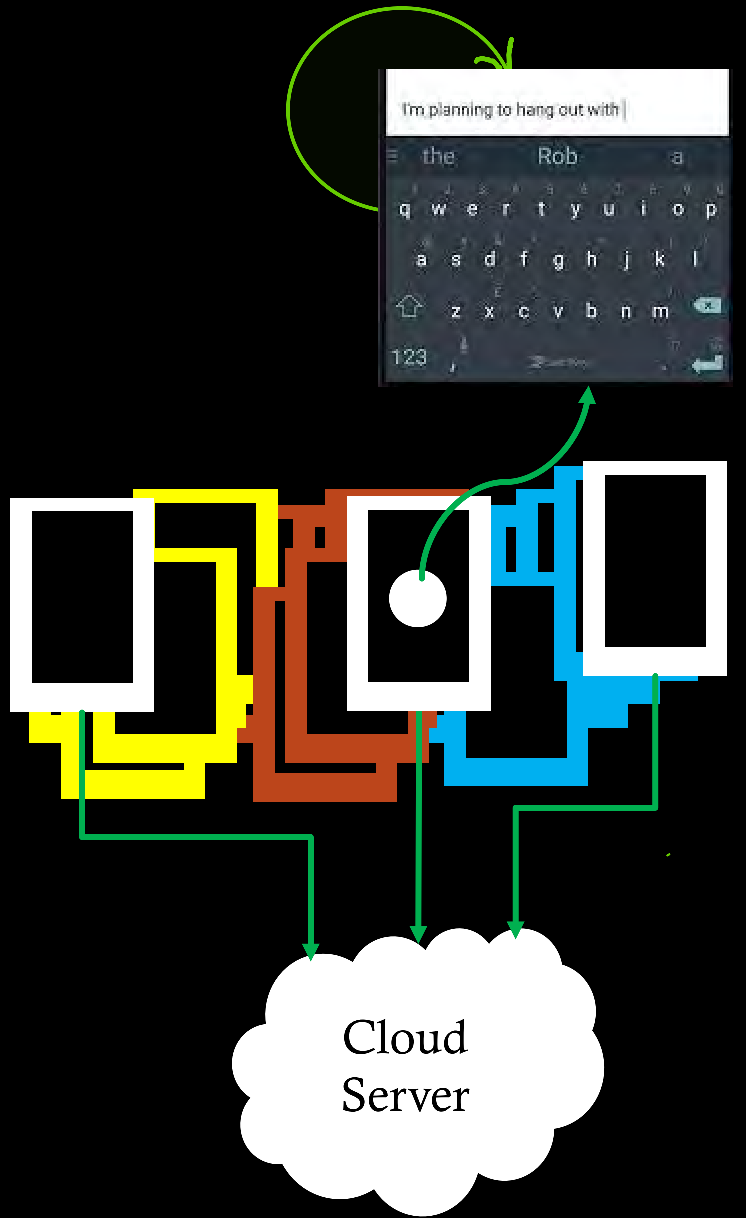
arm

Outline

- Vanilla Federated Learning
  - Concept and Challenges
  - Device Debiasing Algorithm
  - Analysis and experiments
- Personalizing to Edge Device
  - Concept & Challenges
  - Application of Debiasing Scheme
  - Analysis and Experiments
- Customizing to Device Capacity: Challenges

Federated Learning

- Millions of Devices with local models in the System
  - Device SMS = Local Data Collection
    - Private not shared
  - Device model offers local suggestions
    - Receives user feedback locally, updates local model
    - Local model updated over many SMS data (messages).
- Cloud Server
  - Different Devices transmit model update *sporadically*
  - Server fuses received models within a small time-window
    - Transmits to currently active devices
- Active Devices perform model updates



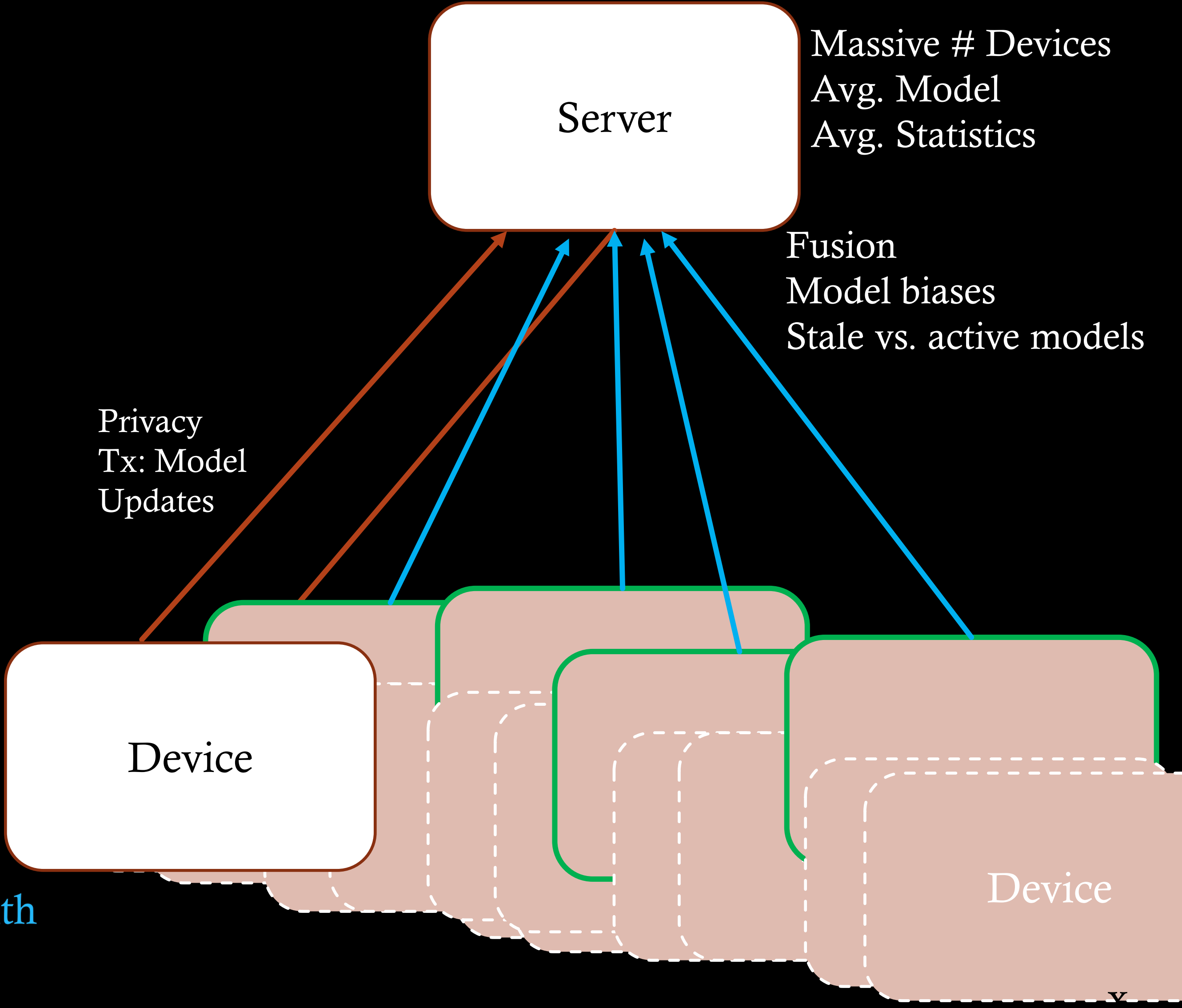
[1] McMahan, H B "Federated Learning: Collaborative Machine Learning without Centralized Training Data." *Google AI Blog*, 6 Apr. 2017.

BOSTON UNIVERSITY

Federated System Constraints

- Privacy vs. Need for lots of training data
  - Device shares only model updates
    - How to balance privacy vs. data?
- Massive # devices.
  - Sporadic device updates
- Device Data Variability.
  - Users have diverse interests/activity
- Device Capacity Variability
  - Samsung S21 Ultra vs. Galaxy A10
- Goals:
  - Accuracy matching training with all device data with minimal cloud/server Tx/RX, and small latency.

CIFAR - 10



Three Problems

- How to training global models matching accuracy of centrally trained (data-shared) models while minimizing communication rounds and bits transmitted?
  - Sporadic device activity, and device data variability (profile and size)
- How to train models on the cloud that can be rapidly personalized to user-specific tasks?
- How to train customized models to meet device capacity specifications?
- Our Solution: Local Debiased Training + Server Model Merging.

Vanilla Federated Learning

BOSTON UNIVERSITY

- Minimize Global Average Loss (m: # Devices, D_k Device k data)

$$\text{Loss}(\theta) = \frac{1}{N}\sum_{k=1}^{m}\sum_{i\in D_k}\text{Loss}(\theta; x_i, y_i)$$

- Local Empirical Loss (available to device k):

$$L_k(\theta; D_k) = \frac{1}{N_k}\sum_{i\in D_k}\text{Loss}(\theta; x_i, y_i)$$

- Challenges
  - Privacy: Device Data not shared.
  - Heterogeneity: imbalanced datasets, not all classes/device
  - Activation: only few among millions of devices.

FedSGD Method

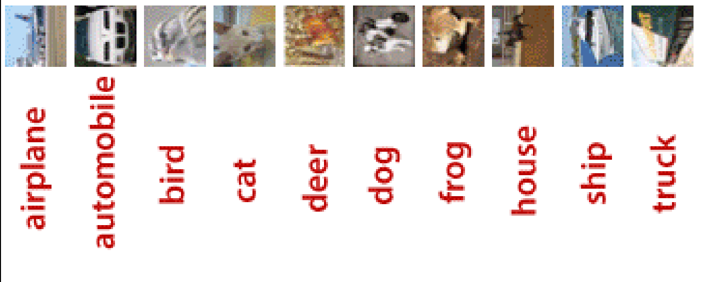
For t=1,2,…
  random devices, $S \subseteq [m]$ and server interact
    server transmits current model, θ_0
    clients $k \in S$ update
    Model update:

$$\theta_k \leftarrow \theta_0 - \eta\nabla L_k(\theta_0; D_k)$$

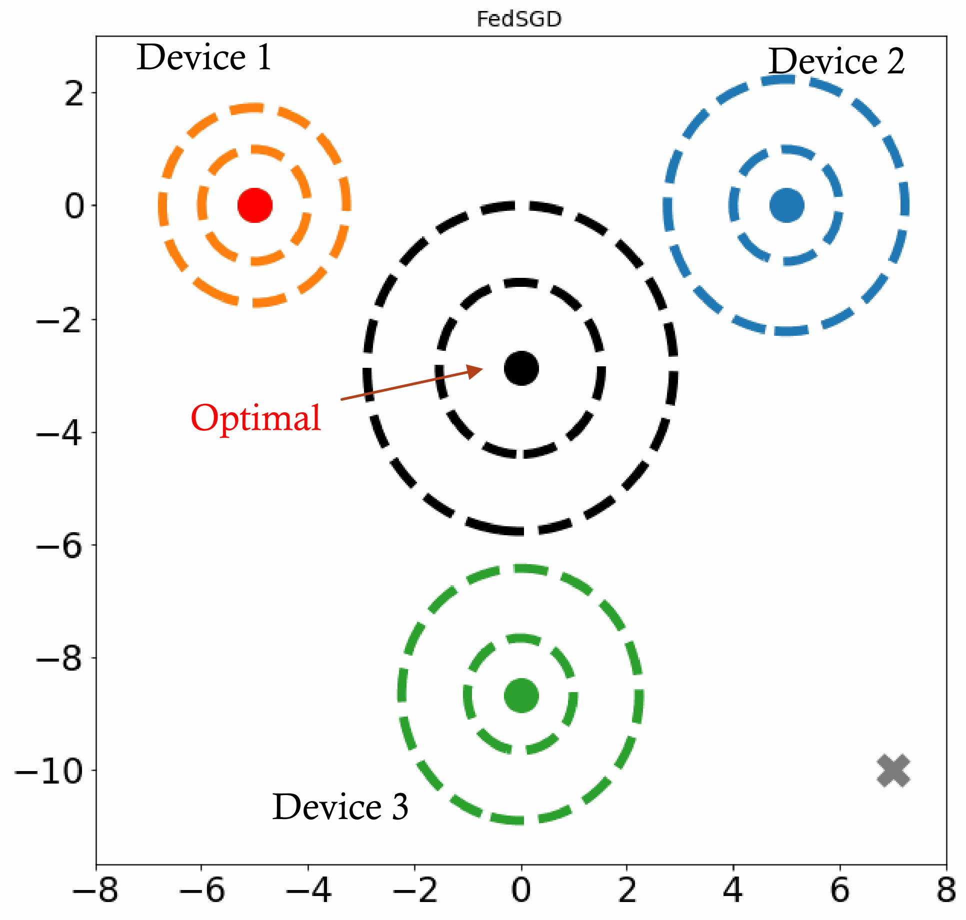
  server receives models and updates:

$$\theta_0 \leftarrow \frac{1}{|S|}\sum_{k\in S}\theta_k$$

FedSGD: High Latency



- Number of Rounds (Latency) for target error δ
  - $T = O(\frac{1}{\delta^2})$
  - Scales inversely with # active clients
- Convergence is slow,
  - # Communication rounds and latency is high



Vanilla ``SGD'' Approach

For t=1,2,…

active devices at time t update cloud model (one gradient step)

$$\theta_k \leftarrow \theta_0 - \eta \nabla L_k(\theta_0; D_k)$$

server merges active devices:

$$\theta_0 \leftarrow \frac{1}{|S|} \sum_{k \in S} \theta_k$$

Federated Averaging and FedProx

- Let device do some of the ``heavy lifting,'' by taking more gradient steps (optimize local loss more)
  - Goal: fewer comms, and low latency.
- Exhibits poor convergence even in convex cases
  - Sporadic activation & data heterogeneity.
- Dilemma: more # grad steps – Bias; few grad: large latency.
  - Introduces #gradient steps as a hyperparameter.

Fed Avg Approach (FedProx = FedAvg + quadratic reg)

For t=1,2,…

active devices perform many gradient steps starting with cloud model

set $\theta' \leftarrow \theta_0$

for i=1,2,…K, do for each active device k:

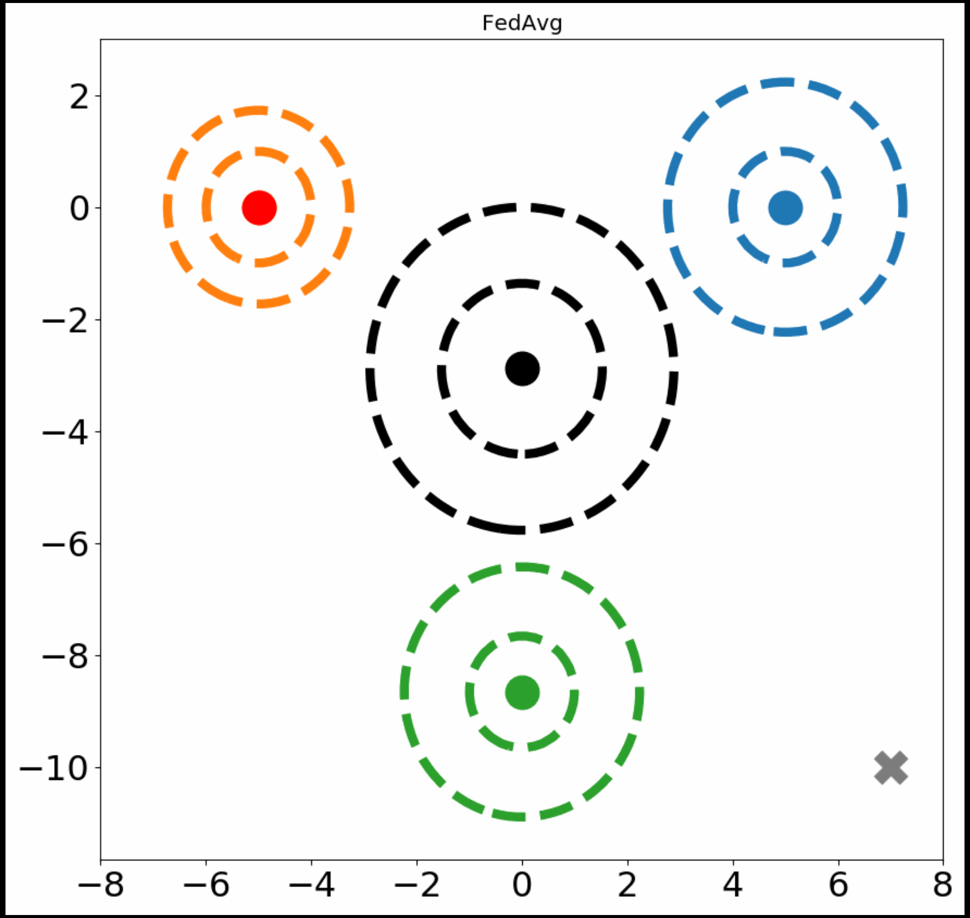
$$\theta_k \leftarrow \theta' - \eta \nabla L_k(\theta'; D_k)$$

$$\theta' \leftarrow \theta_k$$

minimize $L_k(\theta) + \frac{\alpha}{2}\|\theta - \theta_0\|^2$

server merges active devices:

$$\theta_0 \leftarrow \frac{1}{|S|}\sum_{k \in S} \theta_k$$



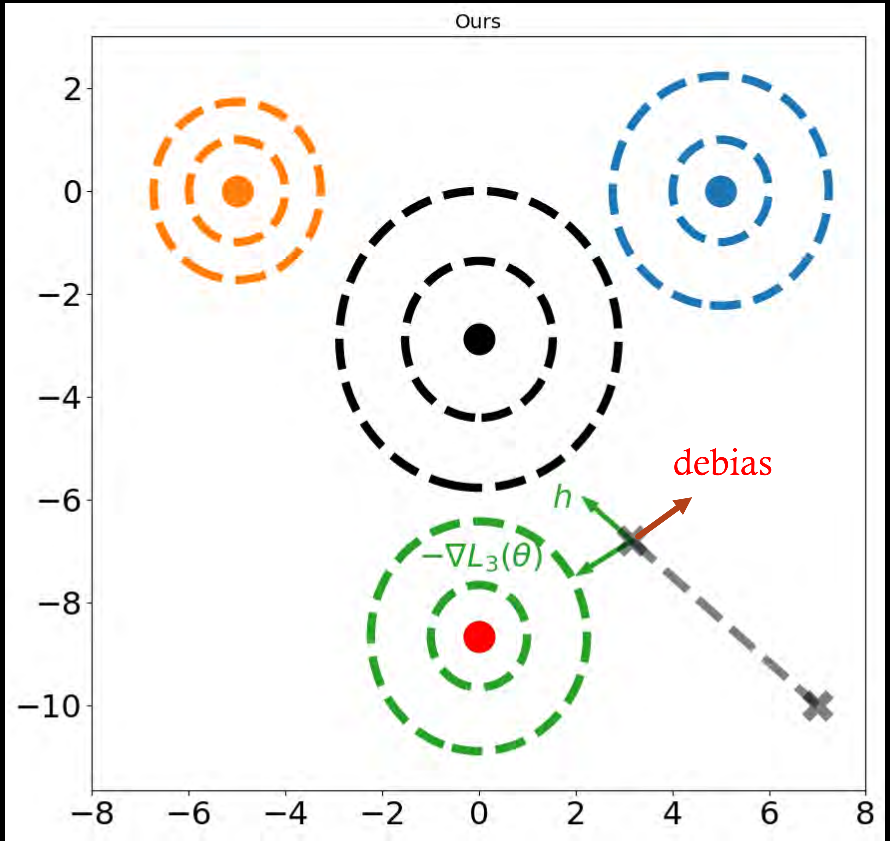
Proposed Scheme: Debiasing Device Model

BOSTON UNIVERSITY

- De-biasing local updates to device dataset
  - Goal: fewer comms, and low latency.
  - FedAvg/FedProx: $\boxed{\text{minimize} \; L_k(\theta) + \frac{\alpha}{2}\|\theta - \theta_0\|^2}$

    Steps in Biased direction
    $-\nabla L_k(\theta_0)$
  - Suppose, oracle provides the correct direction, $h := -\frac{1}{m}\sum_{k\in[m]} \nabla L_k(\theta_0)$
    - This is the global gradient (want it to be zero)!!
    - Fake Device Loss: $L_k(\theta) - \langle \nabla L_k(\theta_0), \theta - \theta_0\rangle + \langle h, \theta - \theta_0\rangle + \frac{\alpha}{2}\|\theta - \theta_0\|^2$
      - subtract biased gradient, add oracle gradient?
  - What is the impact?
    - First step results in: $\theta_k \leftarrow \theta_0 + \eta h$

Biased direction cancelled
Correct direction substituted



Debiasing Device Model Update: All active

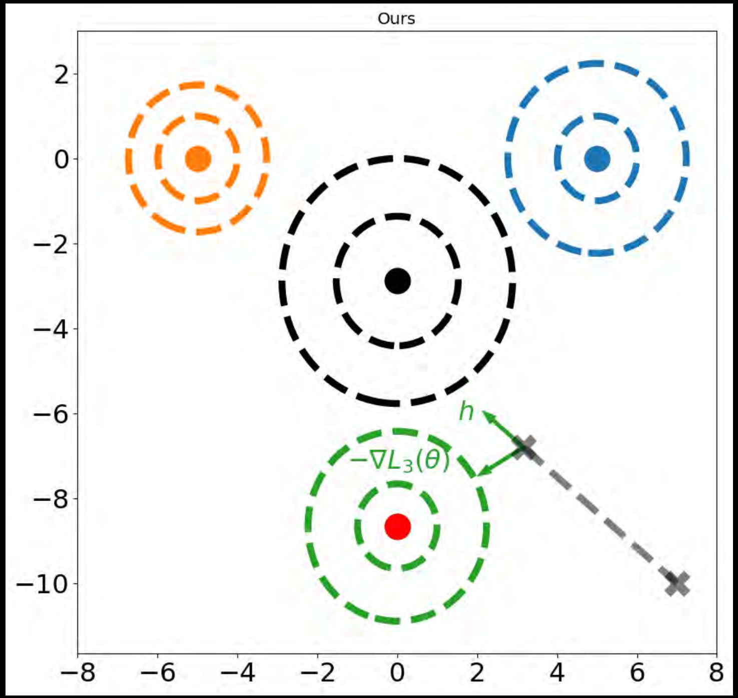
- Fake Obj: minimize $L_k(\theta) - \langle \nabla L_k(\theta_0), \theta - \theta_0 \rangle + \langle h, \theta - \theta_0 \rangle + \frac{\alpha}{2}\|\theta - \theta_0\|^2$

$$-\frac{1}{m}\sum_{k\in[m]} \nabla L_k(\theta_0)$$

- Correct direction unavailable
  - Server has only local models
    - Loss functions are private
  - Sporadic comm activity
    - Server cannot sync all devices

$$h' := -\frac{1}{m}\sum_{k\in[m]} \nabla L_k(\theta_k)$$

Last Tx
device model



- Leverage the last Tx device model
  - Available at cloud server
  - How to circumvent gradient Tx?

Debiasing Device Model Update

- Proposal: Device optimizes

$$\text{minimize}\quad L_k(\theta) - \langle \nabla L_k(\theta_k), \theta - \theta_0 \rangle + \langle h', \theta - \theta_0 \rangle + \frac{\alpha}{2}\|\theta - \theta_0\|^2$$

$$h' = -\frac{1}{m}\sum_{k \in [m]} \nabla L_k(\theta_k)$$

- Device Tx Updated model
  - How does device compute avg approximate gradient?
    - It does not have to!
- Server
  - How should server update?
  - What should server Tx?
    - Model average plus approx. grad!

Server Tx

$$L_k(\theta) - \langle \nabla L_k(\theta_k), \theta - \theta_0 \rangle + \frac{\alpha}{2}\|\theta - \overbrace{\theta_0 + \frac{1}{\alpha}h'}^{\theta_0'}\|^2$$

Device Optima θ_k^+

$$\nabla L_k(\theta_k^+) = \nabla L_k(\theta_k) - \alpha(\theta_k^+ - \theta_0')$$

Linear Eqn

Devices Tx update θ_k^+

Server can update locally!!

Diff in grad = diff in models

(without needing gradient Tx)

Dynamic Debiasing Federated Learning (DyDFL) with Sporadic activation

BOSTON UNIVERSITY

Dynamic Regularization Approach

For t=1,2,…

Server:

Rx: model from active devices: θ_k

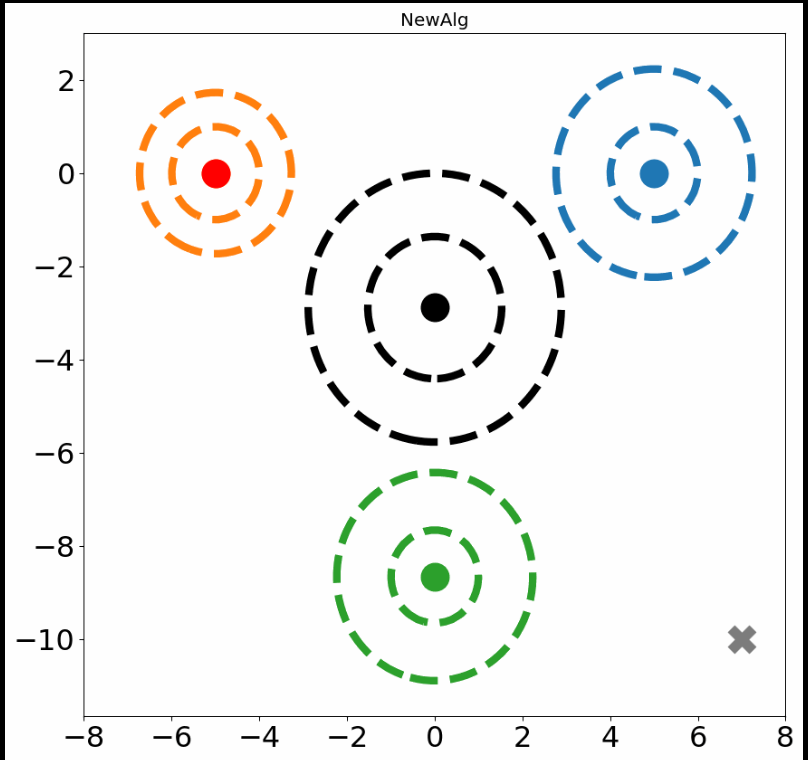
updates state locally: $h' \leftarrow h' - \frac{\alpha}{m} \sum_{k \in S} (\theta_k - \theta_0)$

Model update: $\theta_0 \leftarrow \frac{1}{|S|} \sum_{k \in S} \theta_k$

Tx: $\theta' \leftarrow \theta_0 - \frac{1}{\alpha} h'$

Models: Update model by optimizing dynamic loss

Update local state (gradient).



Does it work? Overcome source bias and converge? Yes!

Analysis of DyDFL

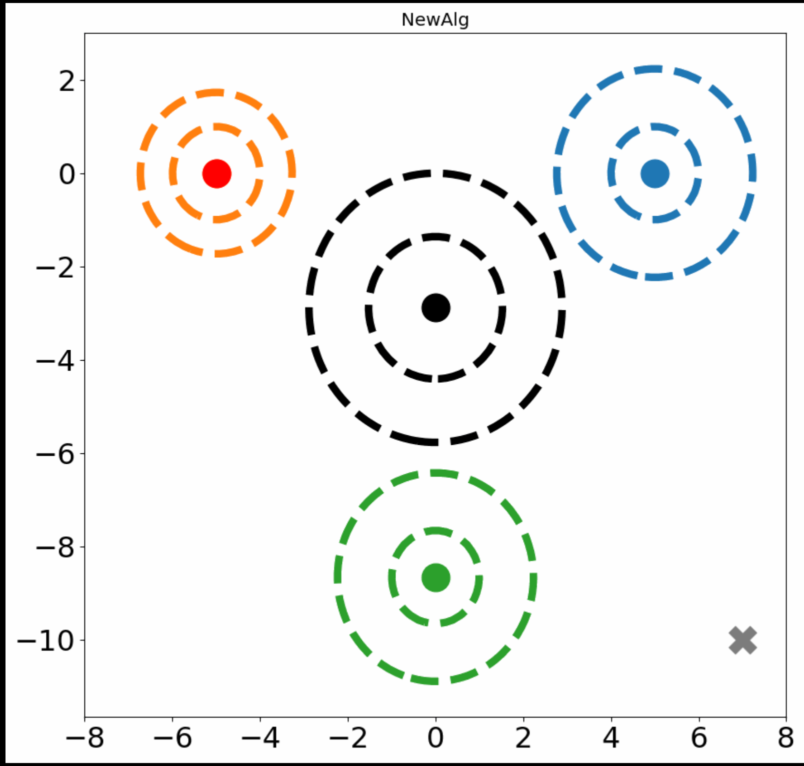
- Convex Case
  - Number of Rounds (Latency) for target error δ

    $$T = O(\sqrt{\frac{m}{S}}\frac{1}{\delta})$$

    - Rapid Convergence and Low Latency
    - Dependence on active participation is ``optimal''
  - FedSGD: $T = O(\frac{1}{\delta^2})$
    - Slow and high latency.
- Non-Convex: # rounds for reaching stationary point

  $$T = O(\frac{m}{S}\frac{1}{\delta})$$

  - State-of-art over prior works
- Bits communicated is same as FedAvg/round (cost: extra local state).
- Minimal hyperparameter tuning



BOSTON UNIVERSITY

Theory & Comparison to Prior Schemes

Method	Partial Participation	Communication per Round	Heterogeneity Assumption	Device Optimization	Extra States
Ours (DyDFL)	Yes	Once	Arbitrary	Exact/SGD	Local and server
FedAvg	Yes	Once	Bounded Gradients	SGD Steps	No
FedProx	Yes	Once	Bounded Gradients	SGD Steps	No
SCAFFOLD	Yes	2X	Arbitrary	SGD Steps	Local and server
FedSplit, FSVRG FedPD, DANE	No	-	-	-	-

FedAvg: McMahan AISTATS 2017.
FEDPROX: Li, MLSys 2020, 2020a.
SCAFFOLD: Karimireddy ICML 2020.
Fedsplit: Pathak R NueRIPS 2020.
FSVRG: Konečný arXiv, 2016.
FEDPD: Zhang arXiv, 2020.
DANE Shamir O, ICML 2014

Experiment Setup

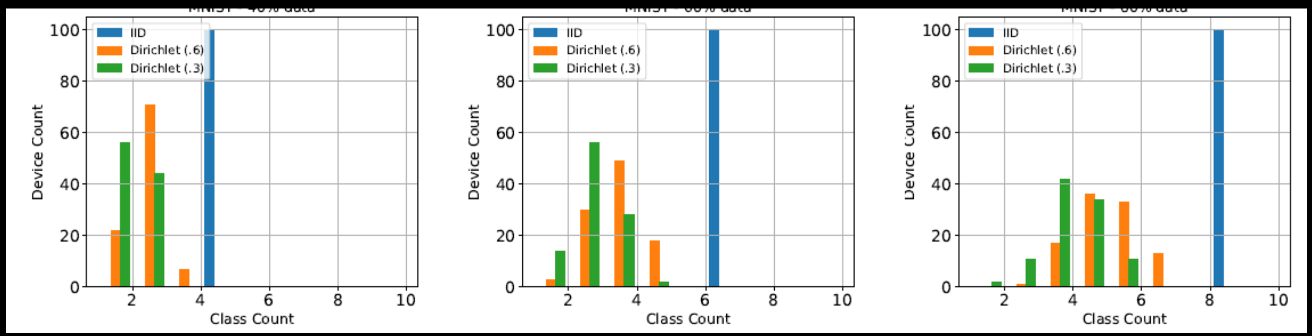
- Datasets
  - Vision:
    - MNIST, CIFAR-10, CIFAR-100, E-MNIST
  - Language:
    - Shakespeare (character prediction)
- Architecture
  - Vision: 2 Conv layers, 2 Fully connected layers
  - Language: Stacked LSTMs

Dataset	Train Samples Amount	Test Samples Amount
CIFAR-10	50000	10000
CIFAR-100	50000	10000
MNIST	60000	10000
EMNIST-L	48000	8000
Shakespeare	200000	40000

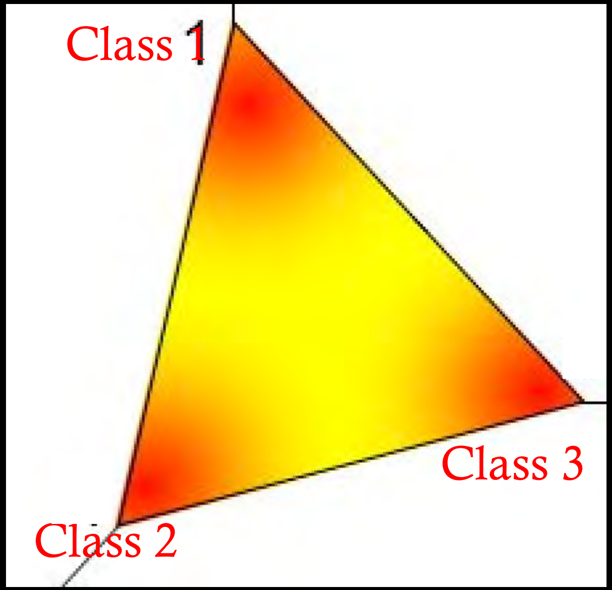
BOSTON UNIVERSITY

Experiment: Evaluation Criteria

- Datasets
  - Vision:
    - MNIST, CIFAR-10, CIFAR-100, E-MNIST
  - Language:
    - Shakespeare (character prediction)
- Architecture
  - Vision: 2 Conv layers, 2 Fully connected layers
  - Language: Stacked LSTMs
- Evaluation Criteria:
  - # Comm rounds to realize target accuracy
    - Data heterogeneity (Dirichlet)
    - Scaling with # Devices
    - Different levels of activity



Dirichlet: 3 classes

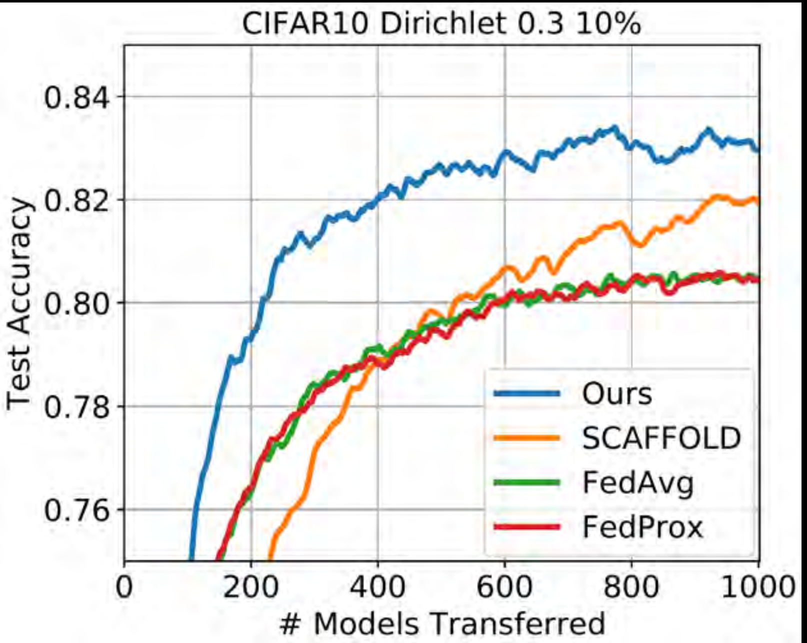


BOSTON UNIVERSITY

Data Heterogeneity (CIFAR-10)

100 Devices, 10% activation, IID, @82.3% Accuracy			
Method	Com. Cost	Gain	# Rounds
Ours (DyDFL)	240		
SCAFFOLD	512	2.1X	
FedAvg	994	4.1X	
FedProx	825	3.4X	

100 Devices, 10% activation, non IID, @80.7%			
Method	Com. Cost	Gain	# Rounds
Ours (DyDFL)	232		
SCAFFOLD	594	2.6X	
FedAvg	863	3.7X	
FedProx	930	4.0X	



Key Insights: 10% activity level

- **DyDFL is agnostic to data heterogeneity.**
- **Achieves fully centralized accuracy (85%)**
- **DyDFL outperforms state-of-art**
- **Similar results for other datasets**

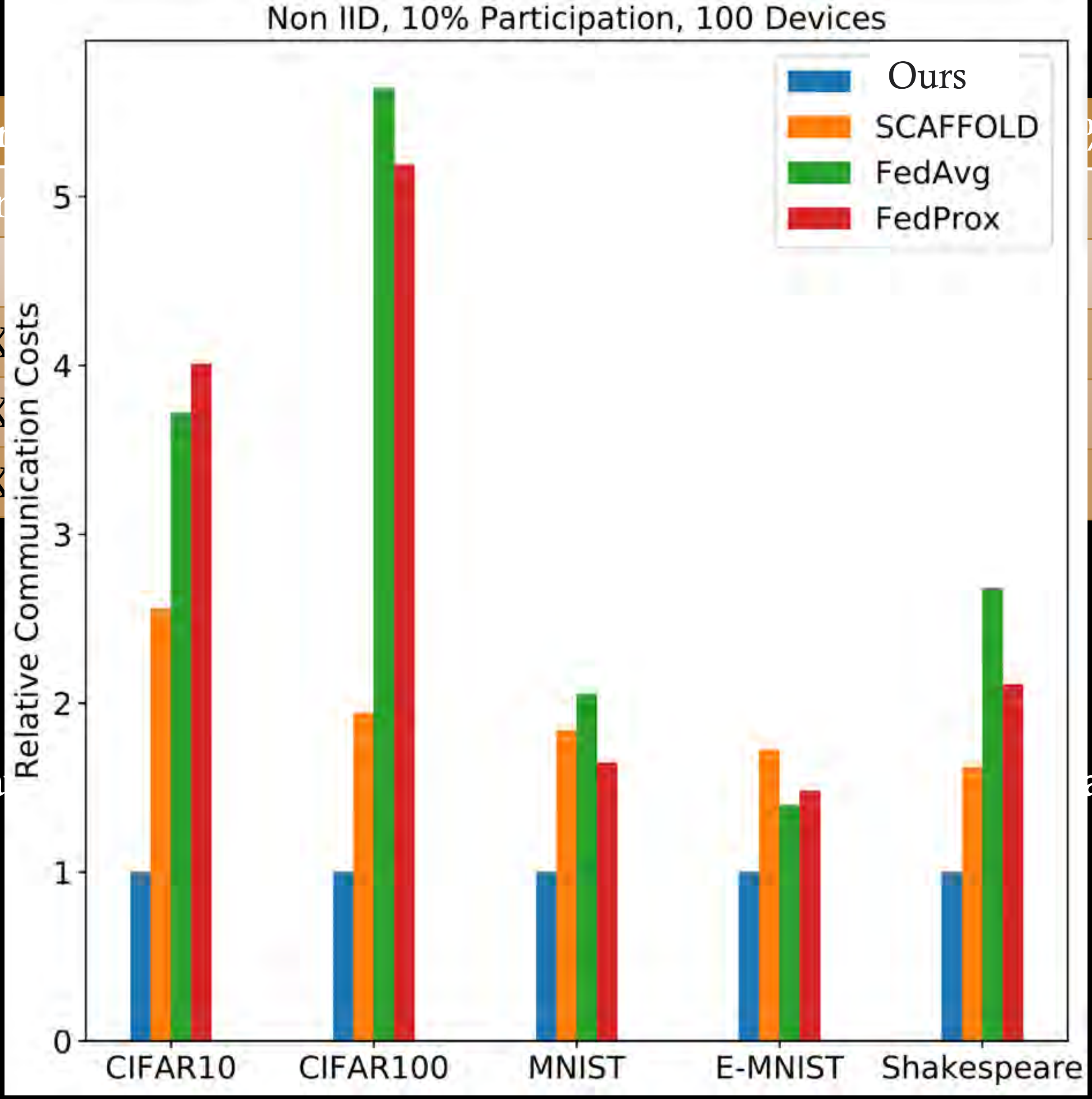
No Loss due to privacy

BOSTON UNIVERSITY

The whole works

CIFAR-10

1000 Devices, 10% activation, n		
Method	Com. Cost	Gai
Ours	350	
SCAFFOLD	2138	6.1X
FedAvg	1962	5.6X
FedProx	1517	4.3X



CIFAR-100

% activation, non IID, @40%	
Gain	
4.9X	
4.1X	
3.8X	

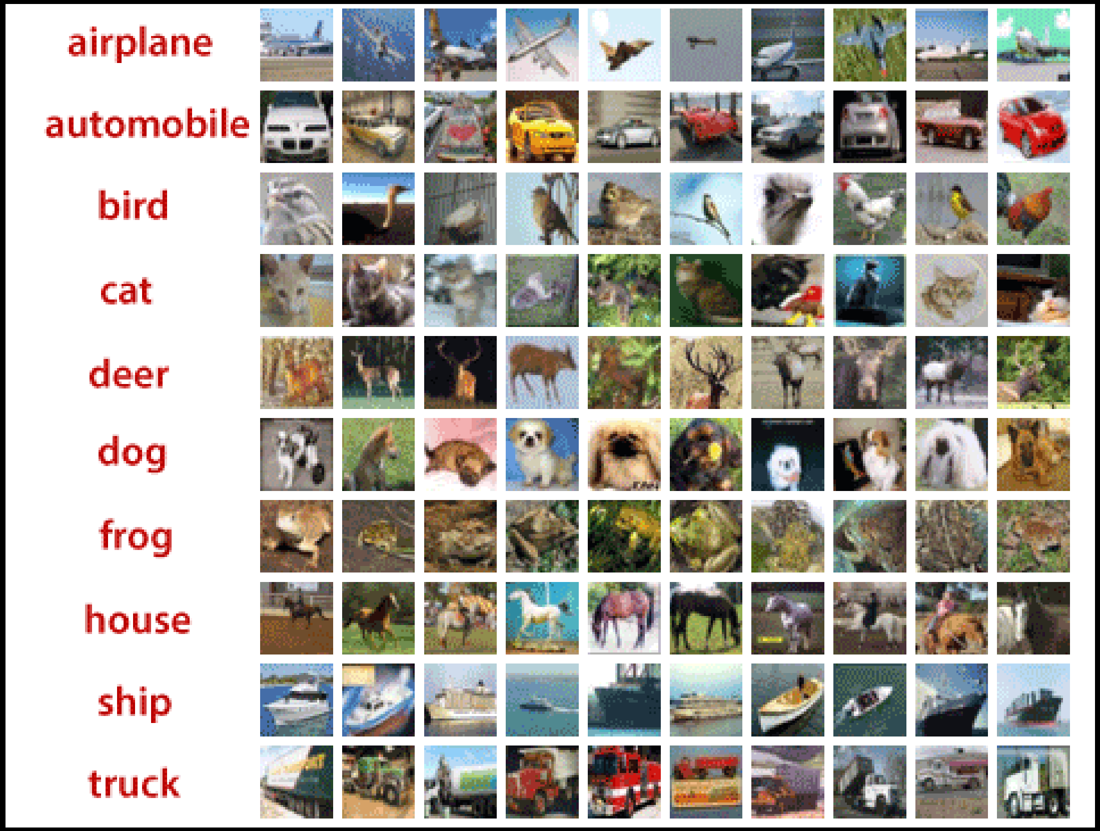
Our method leads to high sa ... and data heterogeneity.

BOSTON UNIVERSITY

Customizing Federated Learning to the Edge

- Objective: Average Personalization Loss (APL)
  - m Devices, kth has dataset D_k, personal objective L_k
    - Example: Device k user ~ has airplane and auto data
      - Interested in airplane type classification, but data is insufficient.
      - less interested in autos.
  - Key Challenges:
    - Global average loss - no longer good objective
    - Global universal model - poor performance on user objective
    - Privacy is even more important, but limited local data means interaction
      - finer-grained airplane classification requires training on larger dataset. Device has a small sample. Privacy implies we cannot do this selectively with a partial subset.

CIFAR - 100



Customizing Federated Learning to the Edge

- Objective: Personalized Device Loss (PDL)
  - m Devices, kth has dataset D_k, personal objective PDL_k

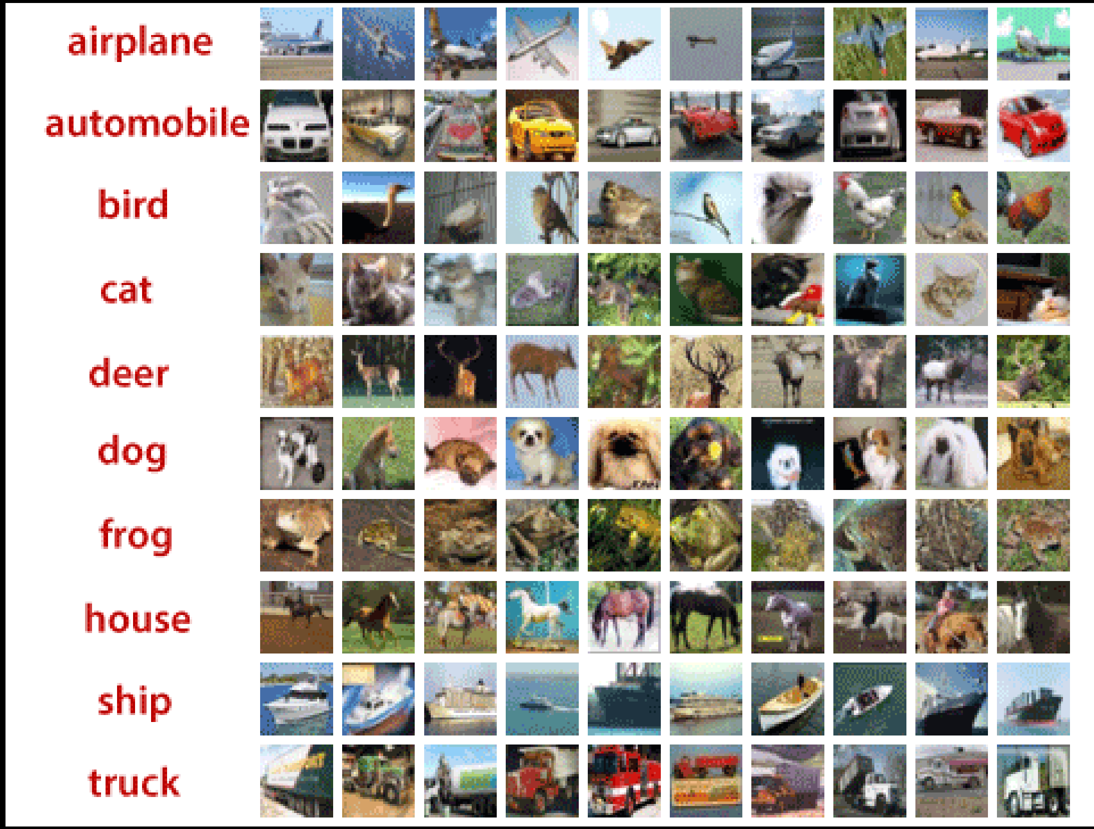
$$PDL_k(\theta_k) = \frac{1}{N_k} \sum_{c \in C} \sum_{\substack{i \in D_k \\ y_i = c}} w_c^k \text{Loss}(\theta_k; x_i, y_i)$$

  - Loss weighted by user interest
  - Model across devices can be different

- Average Personalization Loss (APL):

$$\text{APL} = \frac{1}{m} \sum_{k=1}^{m} PDL_k(\theta_k)$$

CIFAR - 100



How to Benefit from Federated Learning

- Average Personalization Loss: $$\text{APL} = \frac{1}{m}\sum_{k=1}^{m} PDL_k(\theta_k)$$
    - PDL: device personal loss
  - Global Optimization decouples into local device minimization problems!!

- Key Idea: Common global model ~ locally adaptable
  - Rapidly Tunable Network:
    - Learn a global model for classification, which can be further fine-tuned
  - Universal Representation Network
    - Learn a good global representation (metric) so that task predictors

BOSTON UNIVERSITY

Common Global Model with Rapid Fine-Tuning

- APL Decouples $$\text{APL} = \frac{1}{m}\sum_{k=1}^{m} PDL_k(\theta_k)$$
  - Rapidly Tunable Network.
    - Allow small change in network weights
    - Common model, and locally adaptable with fine-tuning T_k

$$PDL_k(\theta_k) = \frac{1}{N_k}\sum_{c\in C}\sum_{i,y_i=c} w_c^k \text{Loss}(\theta_k; x_i, y_i)$$
$$= \frac{1}{N_k}\sum_{c\in C}\sum_{i,y_i=c} w_c^k \text{Loss}(T_k(\theta); x_i, y_i)$$

- APL Minimization:

$$\text{APL} = \frac{1}{m}\sum_{k=1}^{m} PDL_k(T_k(\theta))$$

Device dependent tranformation

- Global Minimization Problem: Can plug-in our DyDFL

Common Feature Representation

BOSTON UNIVERSITY

- APL Decouples $$\text{APL} = \frac{1}{m}\sum_{k=1}^{m} PDL_k(\theta_k)$$

$$PDL_k(\theta) = -\frac{1}{N_k}\sum_{c\in C}\sum_{i} w_c^k \log(p_k(y_i = c \mid \theta(x_i)))$$

Device dependent posterior

  - Universal Representation
    - Induce a metric on feature space ([Protonet'17])
    - Can train classifier or nearest neighbor rule

- APL Minimization:

$$\text{APL} = \frac{1}{m}\sum_{k=1}^{m} PDL_k(\theta)$$

  - Global Minimization Problem

Dynamic Regularization Approach

For t=1,2,…
Server:
Rx: model θ_k from active devices in S:
updates state locally: $h' \leftarrow h' - \frac{\alpha}{m}\sum_{k\in S}(\theta_k - \theta_0)$
Model update: $\theta_0 \leftarrow \frac{1}{|S|}\sum_{k\in S}\theta_k$
Tx: $\theta' \leftarrow \theta_0 - \frac{1}{\alpha}h'$
Models: Update model with (PDL_k)
Update local state (gradient).

Experiment Setup

- Datasets and Network
  - Vision: CIFAR-10, CIFAR-100
  - Architecture: 2 Conv layers, 2 Fully connected layers

- Evaluation Criteria: # Rounds to achieve Target Accuracy
  - APL and Lowest Personalization Loss (LPL)
  - Sparsely (k classes/device) chosen classes uniformly at random
  - Each device randomly permutes class index (PCI).
    - Class index, j, in device k does not correspond to index j in another device.
    - Example: Airplane is indexed as 1 in device 1 but may be any other number in device k
    - Enhances Privacy.

CIFAR - 10

CIFAR10 – Average Personalization

Sparse with correct class association @91.6% Test Acc.			
Method	Rounds	Gain	
Ours (metric)	152		→
Ours (fine-tuning)	242	1.6X	→
FedAvg (metric)	334	2.2X	→
[a,b] (fine-tuning)	815	5.4X	→

Sparse Random Class indices/device @87.9% Test Acc.			
Method	Rounds	Gain	
Ours (metric)	73		→
Ours (fine-tuning)	323	4.4X	→
FedAvg (metric)	95	1.3X	→
[a,b] (fine-tuning)	792	10.8X	→

Ours (metric) has consistently high savings.

Metric adaptation is robust to label permutation.

Ours is better than vanilla FedAvg achieving SOTA.

[a] Fallah, A., Mokhtari, A., & Ozdaglar, A. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. *Advances in Neural Information Processing Systems, 33, 2020*
[b] Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. arXiv preprint arXiv:1802.07876, 2018.

BOSTON UNIVERSITY

CIFAR10 – Worst-Case Personalization Accuracy

Sparse with correct class association @79% accuracy			
Method	Rounds	Gain	
Ours (Metric)	211		→
Ours (fine-tuning)	482	2.3X	→
FedAvg (metric)	512	2.4X	→
[a,b] fine-tuning	312	1.5X	→

Sparse Randomly Permuted Class indices @69%			
Method	Rounds	Gain	
Ours (Metric)	100		→
Ours (fine-tuning)	250	2.5X	→
FedAvg (metric)	166	1.7X	→
[a,b] (fine-tuning)	710	7.1X	→

Ours with metric based adaptation leads to high savings.

Metric based adaptation is robust to label permutation.

Our (metric or fine-tuning) improves the lowest level personalization.

[a] Fallah, A., Mokhtari, A., & Ozdaglar, A. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. *Advances in Neural Information Processing Systems, 33, 2020*
[b] Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. arXiv preprint arXiv:1802.07876, 2018.

CIFAR-100: Average Personalization

BOSTON UNIVERSITY

CIFAR100, Sparse with Correct Class Association			
Test Average Personalization 89.1%			
Method	Com. Cost	Gain	
Ours (metric)	133		→
Ours (fine-tuning)	255	1.9X	→
FedAvg (metric)	383	2.9X	→
[a,b] (fine-tuning)	961	7.2X	→

CIFAR100, Sparse with Randomly Permuted Class Indices			
Test Average Personalization 89.5%			
Method	Com. Cost	Gain	
Ours (metric)	156		→
Ours (fine-tuning)	849	5.4X	→
FedAvg (metric)	390	2.5X	→
[a,b] (fine-tuning)	>1000	>6.4X	→

[a] Fallah, A., Mokhtari, A., & Ozdaglar, A. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. *Advances in Neural Information Processing Systems, 33, 2020*

[b] Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. arXiv preprint arXiv:1802.07876, 2018.

CIFAR-100: Worst-Case Personalization

CIFAR100, Sparse with Correct Class Association			
Test Lowest Personalization 75.0%			
Method	Com. Cost	Gain	
Ours (metric)	254		
Ours (fine-tuning)	949	3.7X	
FedAvg (metric)	714	2.8X	
[a,b] (fine-tuning)	>1000	>3.9X	

CIFAR100, Sparse with Random Class Association			
Test Lowest Personalization 73.0%			
Method	Com. Cost	Gain	
Ours (metric)	192		
Ours (fine-tuning)	721	3.8X	
FedAvg (metric)	692	3.6X	
[a,b] (fine-tuning)	>1000	>5.2X	

[a] Fallah, A., Mokhtari, A., & Ozdaglar, A. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. *Advances in Neural Information Processing Systems, 33, 2020*
[b] Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. arXiv preprint arXiv:1802.07876, 2018.

Conclusion

- Federated Learning
  - Privacy, Data heterogeneity, Sporadic device activation, millions of devices
  - Device bias is a significant issue
  - Proposed Debiasing algorithm, which allows device to fully optimizing local objective
    - Requires no hyperparameter tuning on epochs etc.
  - Theory and experiments demonstrate significant computational and communication gains.
- Customization to edge device
  - User objectives, but can also include device capacity (upcoming work)